ALADIN User's Manual

Daniel Egret[1,2], François Bonnarel[1], Pierre Fernique[1], Thomas Boch[1], Olivier Bienaymé[1]

[1] CDS, UMR 7550, Observatoire de Strasbourg, 11 rue de l'Université, 67000 Strasbourg, France
[2] present address: Observatoire de Paris
Email: question@simbad.u-strasbg.fr

February 16, 2004– **Aladin Manual 2nd release – describing Aladin version 2.**

Abstract. ALADIN is an interactive sky atlas developed and maintained by the Centre de Données astronomiques de Strasbourg (CDS) for the identification of astronomical sources through visual analysis of reference sky images.

Contents

1 Introduction **3**
1.1 The ALADIN interactive sky atlas 3
1.2 The CDS 3

2 The user interfaces **3**
2.1 Aladin previewer 3
2.2 Aladin applet 3
2.3 Aladin Standalone 4

3 Using Aladin **4**
3.1 Access 4
3.2 Getting started 4
3.3 The view panels 5
3.4 The plane stack 5
3.4.1 Image Plane 5
3.4.2 Simbad 5
3.4.3 VizieR 5
3.4.4 NED 6
3.4.5 Mission logs and Image Archives . . . 6
3.4.6 Instrument Field of View 6
3.4.7 My Data 6
3.5 Learning more about the astronomical objects . 6
3.5.1 Selecting objects 6
3.5.2 The Measurement window 7
3.6 The tool bar 7
3.6.1 Properties 7
3.6.2 Draw 7
3.6.3 Text, Tags 7
3.6.4 Distance (Dist) 7
3.6.5 RGB colour composition 7
3.6.6 Contour plots 8
3.6.7 Colour map (Histog.) 8
3.6.8 Zoom and image manipulation 8
3.6.9 Scale and coordinate grid 8
3.6.10 Cut-and-paste using the Pad window . . 8
3.7 Help 8

4 More Detailed Features **9**
4.1 Manipulating the plane stack 9
4.2 Interacting with the VizieR database 9
4.3 Shortcut commands for loading data 10
4.4 Coordinates manipulation 10
4.5 Using the SHIFT key for hidden features 10
4.6 URL command lines 10

5 Filters in Aladin **11**
5.1 Introduction 11
5.2 Syntax 11
5.2.1 Invocating columns 11
5.2.2 Comments 12
5.2.3 Arithmetic expressions using UCDs/columns 12
5.2.4 Conditions 12
5.2.5 Constraints 13
5.2.6 Actions 13
5.2.7 Defining a color 13
5.2.8 Symbol shape 13
5.3 Usage in Aladin 14
5.3.1 Creating a filter 14
5.3.2 Modifying a filter 14
5.3.3 Syntax errors 14
5.3.4 Activate/Deactivate a filter 14
5.3.5 Scope of a filter 14
5.3.6 Applying multiple filters 15
5.3.7 Creating a filter in script mode 15
5.3.8 Miscellaneous 15

6 The image database **15**
6.1 Database summary 15
6.2 DSS Images 15
6.3 MAMA/CAI Images 16
6.4 2MASS near-infrared survey 16
6.5 The image server 16
6.6 Image astrometry 16
6.7 Image compression 16

7 The catalogue and survey server 17
7.1 VizieR 17
7.2 Archive previews 17

8 Specific features of the Stand-alone version 17
8.1 Starting the Stand-alone version 17
8.1.1 The Java Virtual Machine 17
8.1.2 The AladinJava Stand-alone package . 17
8.2 Loading personal files with Aladin Standalone . 18
8.2.1 Local images 18
8.2.2 Calibrating Images 18
8.2.3 Local Tables 18
8.3 Saving and Printing 18
8.4 Defining additional servers 19
8.5 Script language 19

9 User feedback 20

10 Acknowledgements 20

A Backus Naur Form of filter syntax 21

1. Introduction

1.1. The ALADIN interactive sky atlas

ALADIN is an interactive sky atlas developed and maintained by the Centre de Données astronomiques de Strasbourg (CDS) for the identification of astronomical sources through visual analysis of reference sky images (Bonnarel et al. 2000).

ALADIN fully benefits from the environment of CDS databases and services (SIMBAD reference database, VizieR catalogue service, etc.), and is designed as a multi-purpose service for use by the professional astronomical community.

ALADIN allows the user to visualize digitized images of any part of the sky, to superimpose entries from the CDS astronomical catalogues and tables, and to interactively access related data and information from SIMBAD, NED, VizieR, or other archives for all known objects in the field.

ALADIN is particularly useful for multi-spectral cross-identifications of astronomical sources, observation preparation and quality control of new data sets (by comparison with standard catalogues covering the same region of sky).

The ALADIN interactive atlas is available in three main modes: a simple previewer, a java applet, and a java standalone application (to be downloaded by the user). ALADIN is publicly accessible on the World-Wide Web at the address: http://aladin.u-strasbg.fr/.

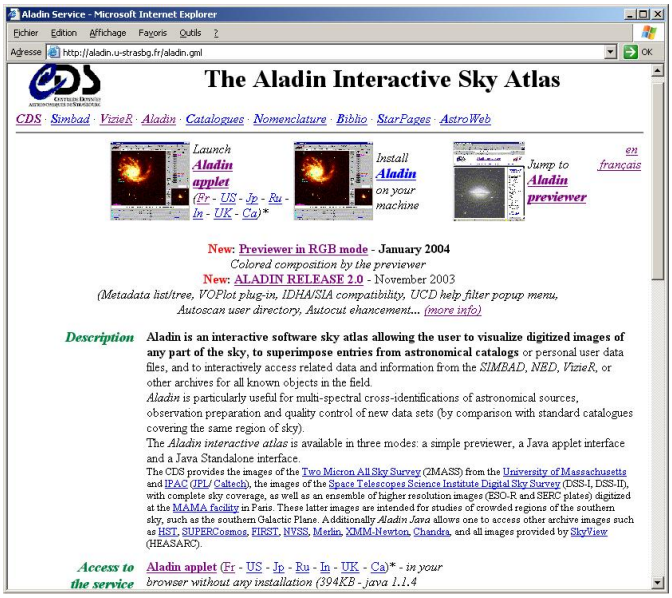
Aladin Service - Microsoft Internet Explorer
The Aladin Interactive Sky Atlas
CDS · Simbad · VizieR · Aladin · Catalogues · Nomenclature · Biblio · StarPages · AstroWeb
Launch Aladin applet (Fr - US - Jp - Ru - In - UK - Ca)*
Install Aladin on your machine
Jump to Aladin previewer
en français
New: Previewer in RGB mode - January 2004
Colored composition by the previewer
New: ALADIN RELEASE 2.0 - November 2003
(Metadata list/tree, VOPlot plug-in, IDHA/SIA compatibility, UCD help filter popup menu, Autoscan user directory, Autocut enhancement... (more info)
Description
Aladin is an interactive software sky atlas allowing the user to visualize digitized images of any part of the sky, to superimpose entries from astronomical catalogs or personal user data files, and to interactively access related data and information from the SIMBAD, NED, VizieR, or other archives for all known objects in the field.
Aladin is particularly useful for multi-spectral cross-identifications of astronomical sources, observation preparation and quality control of new data sets (by comparison with standard catalogues covering the same region of sky).
The Aladin interactive atlas is available in three modes: a simple previewer, a Java applet interface and a Java Standalone interface.
The CDS provides the images of the Two Micron All Sky Survey (2MASS) from the University of Massachusetts and IPAC (IPL/ Caltech), the images of the Space Telescopes Science Institute Digital Sky Survey (DSS-I, DSS-II), with complete sky coverage, as well as an ensemble of higher resolution images (ESO-R and SERC plates) digitized at the MAMA facility in Paris. These latter images are intended for studies of crowded regions of the southern sky, such as the southern Galactic Plane. Additionally Aladin Java allows one to access other archive images such as HST, SUPERCosmos, FIRST, NVSS, Merlin, XMM-Newton, Chandra, and all images provided by SkyView (HEASARC).
Access to the service
Aladin applet (Fr - US - Jp - Ru - In - UK - Ca)* - in your browser without any installation (394KB - java 1.1.4

Fig. 1. Home page of ALADIN at the CDS Web site.

1.2. The CDS

The Centre de Données astronomiques de Strasbourg (CDS) defines, develops, and maintains services to help astronomers find the information they need from the very rapidly increasing wealth of astronomical information.

A detailed description of the CDS on-line services can be found, e.g., in Genova et al. (2000), or at the CDS web site: http://cdsweb.u-strasbg.fr/.

2. The user interfaces

ALADIN is currently available in three main modes: ALADIN previewer; java applet; java stand-alone version.

2.1. Aladin previewer

The ALADIN previewer is a pre-formatted image server providing a compressed image of fixed size, around a given object or position, either in full resolution ($14.1' \times 14.1'$ for the DSS-I used as default reference atlas), or as a global plate view (approximately $6° \times 6°$ according to the survey). When an object name is given, its position is resolved through the SIMBAD name resolver.

Fig. 2. Result page of ALADIN previewer for the galaxy M51. The default image is from DSS-I.

The result page (Fig. 2) gives also access to the other survey images available, and to the full resolution FITS image for download.

This is a direct and simple tool for sky visualization, without the interactive features (catalogue access, overlays...) of the main Java version.

2.2. Aladin applet

The Aladin Java applet is the primary public interface, supporting queries to the CDS image server, overlays from any catalogue or table available at CDS, or from SIMBAD and NED databases, and access to a number of remote archives.

ALADIN applet is available directly through current Web browsers (such as Netscape, Mozilla or IExplorer).

It allows the user to query images, catalogues, data, to manipulate the images by zooming or modifying the dynamics, to access the full records of any sources, to add or draw symbols, vectors, etc.

Access to personal files is not possible from the applet (due to security restrictions of the Java language). However, personnal files accessible through a URL (section 3.4.7) remains available with the Java applet version. These restrictions do not apply to the *stand-alone*. version (see below).

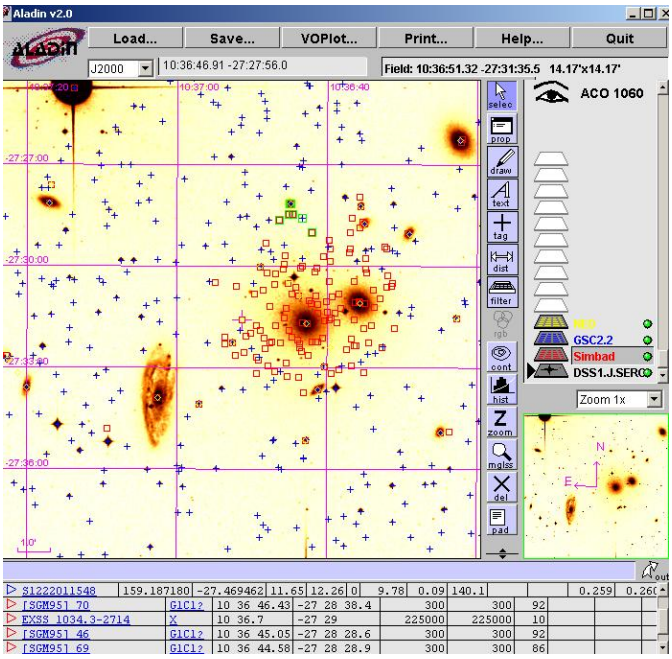

Fig. 3. Example of ALADIN window, with an image centered on the Hydar I cluster of galaxies ACO 1060, and objects from SIMBAD, GSC2.2 and NED marked by symbols.

Aladin Java is compatible from Java release 1.1.2. If your web browser (Explorer, Netscape, Mozilla...) supports java, you support Aladin Java Applet. On some recent browsers you have to install a java plugin machine, then follow your browser documentation.

ALADIN Java interface and tools are developed and maintained by *Pierre Fernique*[1] (CDS, Strasbourg). Contour, filter and data tree tools have been developed by *Thomas Boch*[2] (CDS, Strasbourg) in the framework of the Astrophysical Virtual Observatory (AVO), an EC RTD project 2002-2004.

2.3. Aladin Standalone

ALADIN Standalone is a program that you can download in order to run Aladin Java as a stand-alone application on your local machine. The main advantage is to overcome general security restrictions of on-line applets: you can load and save image/data on your disk, print finding charts, etc.

If you choose this solution, you will have, if this is not yet done, to download the *Aladin Java package*[3].

Instructions concerning the installation of these packages are given in Section 8.

[1] http://cdsweb.u-strasbg.fr/people/pf.html
[2] http://cdsweb.u-strasbg.fr/ boch
[3] http://aladin.u-strasbg.fr/AladinJava?frame=downloading

3. Using Aladin

3.1. Access

The ALADIN home page is available through the CDS Web server at the following address: http://aladin.u-strasbg.fr/

This site provides access to ALADIN documentation, including scientific reports, recent publications, etc.

A list of the mirror sites for the ALADIN applet is also kept up-to-date on the home page. Note that a mirror copy of the applet is available at the NASA/IPAC Extragalactic Database (NED) where it provides access to the NED collection of FITS images featuring extragalactic objects, and another one at the CFHT, where it is used to assist submissions for the Queue Service Observing.

3.2. Getting started

The typical usage scenario starts with a request of the digitized image for an area of the sky defined by its central position or name of central object (to be resolved by SIMBAD). This is done by pushing the Load button which opens the query panel in a new window (Fig.4). The left side of the query panel lists the different image servers (starting with the Aladin image collection at CDS), while the right side lists the data and archive servers. Push the button corresponding to the Aladin server, type in the Target box the name object, push the submit button and the list of all available images for the field will be displayed. The user is expected to select one (or several) of the proposed images, and to push the submit button.

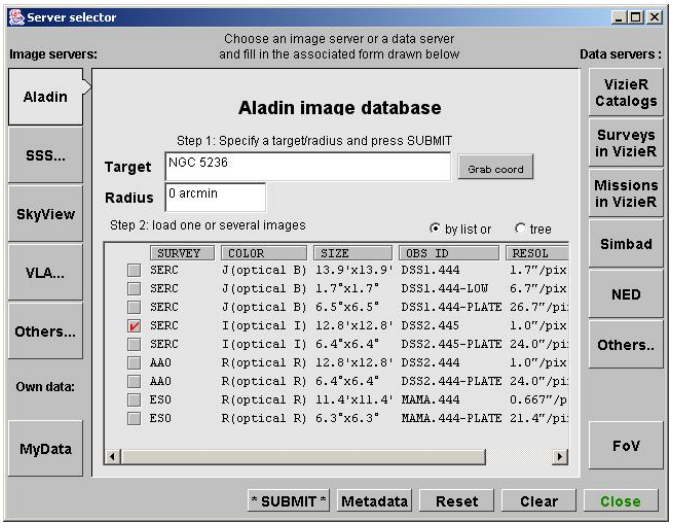

Fig. 4. Example of Images/Data ALADIN Load panel.

Images are extracted, e.g., from Digitized Sky Surveys (DSS) covering the whole sky in several colours (section 6). The size of a sky field is determined by the resolution of the image. For full resolution images it is 14.1′ in the case of DSS-I and 12.9′ for DSS-II. The image server also currently provides DSS images in low resolution (about $1.5° \times 1.5°$), and global views of the original photographic Schmidt plates (about

$6° \times 6°$). Other image surveys, such as the 2 Micron All Sky Survey (2MASS), are also included in the image database.

To abort a query, click on the plane name associated to your query. Remove this plane by clicking the Del button in the tool bar.

3.3. The view panels

The ALADIN interface comprises the following windows and panels (Fig. 5):

- View window: the main window used for displaying the image or catalogue projections.
- Position (and shortcut) window: the small rectangular panel above the View window is used for displaying the current position of the cursor. By clicking on it you can grab the position of the last selected object, or enter a shortcut command (see section 4.3).
- Plane stack: this is the panel at the right of the View window topped by the eye logo. The status of the different planes is specified by icons and colour signals (sections 3.4 and 4.1).
- Tool bar: a set of icons between the View windows and the Plane stack gives access to control functions (3.6).
- Zoom window: the square window at the bottom right corner displays a quick view of the complete field, and can be used to monitor the zoom factor (3.6.8).
- Measurement window: the rectangular panel at the bottom is used for displaying data related to objects selected (with mouse and cursor) in the View window (3.5.2). It can be split into a separate (wider) window.
- Status panel: a small rectangular area between the View window and the Measurement window is mainly used for displaying status messages related to selected objects.

Contextual menus appear when clicking on the right button of the mouse. These menus are different according to the view panels.

3.4. The plane stack

The interface allows the user to stack several information planes related to the same sky field, to superimpose related data from catalogues and databases, and to obtain interactive access to the original data.

The user's eye symbolized by the eye logo sees the projection of all active planes in the View window. A specific icon shows the type of a plane (image, data, graphical element, or filter). The status of a plane is specified by a coloured signal to the right of its name: flashing when the data are being retrieved, green when the data have been successfully retrieved, red when the query failed, orange for warning (e.g. no astrometric reduction).

A black triangle at the left of the plane logo shows which plane is used as the reference projection (see section 4.1 for more details on how to use the plane stack).

You can click on a plane icon to activate or deactivate it (the icon is dark when the plane is active). When clicking on the plane name you select it for the control buttons authorized for

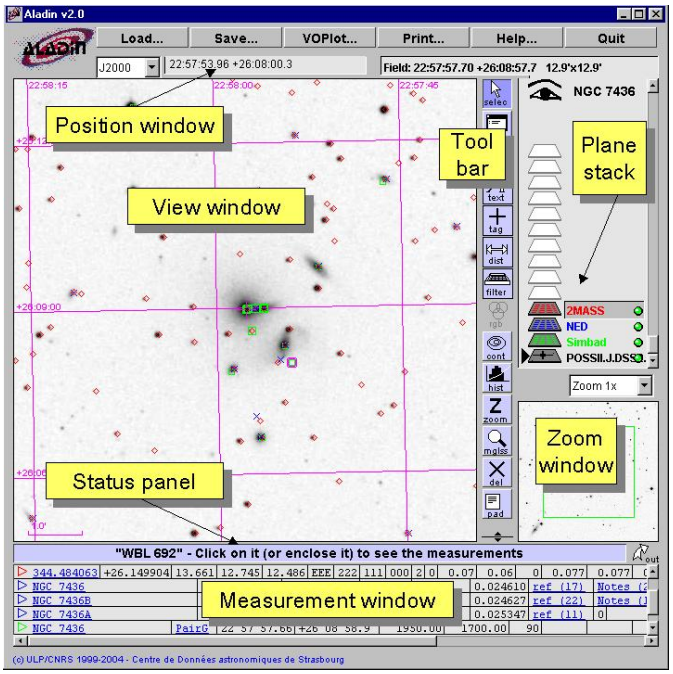

Fig. 5. Layout of the ALADIN interface window.

this plane type (e.g. the Prop. button displaying its properties). Click and drag the plane icons to change the order of display in the stack. The first active image hides all planes underneath.

The possible information planes can be of four different types: image planes (from the image data base or from observatory archives), data planes (from SIMBAD, NED or VIZIER), graphics (labels, tags, Field of Views, etc.), or filters (section 5).

3.4.1. Image Plane

Image pixels from the ALADIN database of digitized photographic plates (DSS-I, MAMA, DSS-II), from the 2MASS survey, from other image servers (SkyView, SuperCosmos), or from distributed image archives (e.g. HST). Functionalities include zooming capabilities, inverse video, modification of the colour table (see 3.6.7, 3.6.8, 6).

3.4.2. Simbad

Information from the SIMBAD database (Wenger et al. 2000); objects referenced in SIMBAD are visualized by colour symbols overlaid on top of the image; the shape and colour of the symbols can be modified on request, and written labels can be added for explicit identification of the objects; *these features are also available for all the other data planes*.

3.4.3. VizieR

Data from the CDS library of catalogues or tables (*VizieR*[4]; Ochsenbein et al. 2000); the user can type a catalogue name, or select the desired catalogue from a preselected list including

[4] http://vizier.u-strasbg.fr/

the major reference catalogues and surveys; the user can alternatively select the catalogues for which entries may be available in the corresponding sky field, using the VIZIER query mechanism by position, catalogue name or keyword; see below 4.2 for more details.

3.4.4. NED

Information from the NED database: objects referenced in the *NASA/IPAC Extragalactic Database*[5] can also be visualized through queries submitted to the NED server at IPAC.

3.4.5. Mission logs and Image Archives

Archive images through mission logs: Hubble Space Telescope (HST), Chandra, VLA/NRAO and FIRST images are currently available, among others. The user should first query the corresponding observation log before retrieving the archived images from the Image, Association(CADC) and similar buttons in the Measurement field (Fig. 6 and 7).

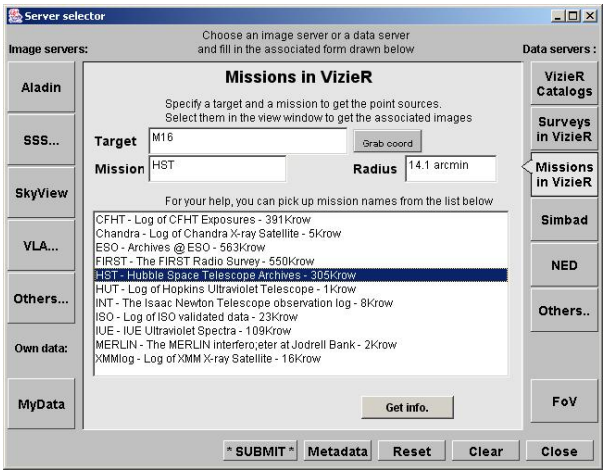
Server selector
Choose an image server or a data server
and fill in the associated form drawn below
Image servers:
Data servers :
Aladin
SSS...
SkyView
VLA...
Others...
Own data:
MyData
Missions in VizieR
Specify a target and a mission to get the point sources.
Select them in the view window to get the associated images
Target
M16
Grab coord
Mission
HST
Radius
14.1 arcmin
For your help, you can pick up mission names from the list below
CFHT - Log of CFHT Exposures - 391Krow
Chandra - Log of Chandra X-ray Satellite - 5Krow
ESO - Archives @ ESO - 563Krow
FIRST - The FIRST Radio Survey - 550Krow
HST - Hubble Space Telescope Archives - 305Krow
HUT - Log of Hopkins Ultraviolet Telescope - 1Krow
INT - The Isaac Newton Telescope observation log - 8Krow
ISO - Log of ISO validated data - 23Krow
IUE - IUE Ultraviolet Spectra - 109Krow
MERLIN - The MERLIN interfero;eter at Jodrell Bank - 2Krow
XMMlog - Log of XMM X-ray Satellite - 16Krow
Get info.
VizieR Catalogs
Surveys in VizieR
Missions in VizieR
Simbad
NED
Others..
FoV
* SUBMIT *
Metadata
Reset
Clear
Close

Fig. 6. Load panel for Mission logs and image archives. An example of data extracted from the HST mission log can be seen in Fig. 7.

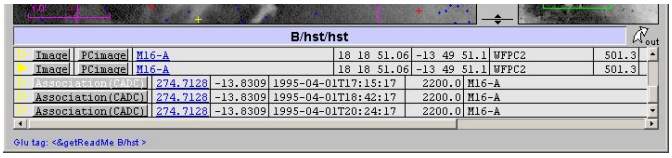

Fig. 7. Querying image archives: the archive log contains pointers to the actual images indicated by Image buttons in the corresponding lines of the Measurement window. See Fig. 8 for an example of HST image display.

[5] http://nedwww.ipac.caltech.edu/

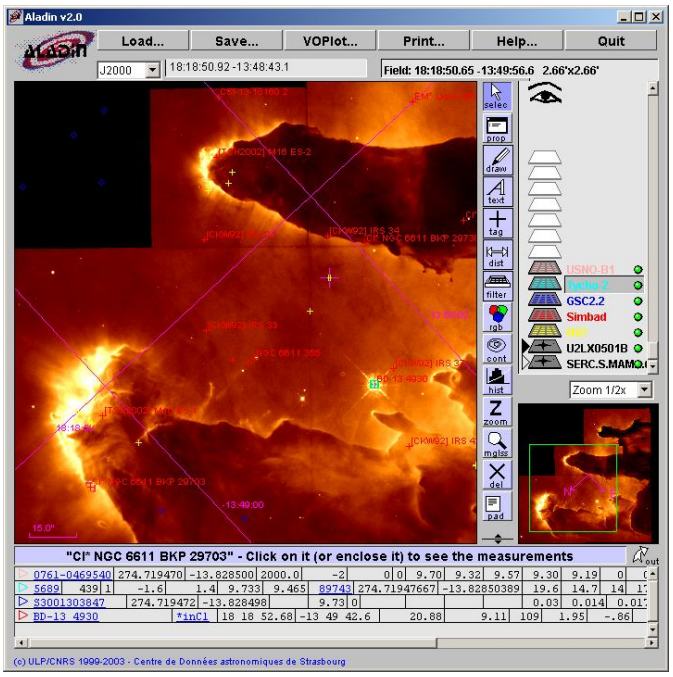

Fig. 8. Example of ALADIN display of a famous image from the HST archive (WFPC2) featuring the Eagle Nebula. Objects present in Simbad, GSC, and USNO B1.0 are flagged with different symbols. Field size is 2.6 arcmin (full image, right) and 1.6 arcmin (left).

3.4.6. Instrument Field of View

It is possible to overlay the field of view (FOV) of some major instruments (presently: CFHT cameras, XMM-Newton and ISO instruments, HST–WFPC2 camera). Rotation of the field, when applicable, can be monitored through the Prop. tool.

3.4.7. My Data

Local, user data files can also be overlaid: just specify an URL and push the submit button. The stand-alone version allows to load files directly from directories. (see Section 8.2 below).

3.5. Learning more about the astronomical objects

3.5.1. Selecting objects

For information planes related to data bases (SIMBAD, VIZIER, NED) links are provided to the original data. This is done in the following way: when selecting an object on the image, with mouse and cursor, summary information is displayed in the Measurement panel on the bottom of the ALADIN window (section 3.5.2); the data line includes links to the original data that can be activated. The retrieved information will appear in a separate window on the Internet browser.

It is also possible to select with the mouse and cursor all objects in a rectangular area: the corresponding objects are listed in the Measurement window; the colour of the triangle at the left of each line recalls the data plane to which it refers.

To select all the objects of a data plane, use the contextual popup menu related to the plane stack.

At any moment the position of the cursor is translated in terms of right ascension and declination on the sky and visualized in the Position window (see also 4.4).

3.5.2. The Measurement window

When objects are selected as explained above, the data for these objects corresponding to the active data planes are listed in the Measurement window; on the bottom of the ALADIN window. This list includes basic information (name, position and, when applicable, number of bibliographical references) and active links to the original catalogue or database. Clicking on this link will open a separate browse window with the related data or bibliography from SIMBAD, NED, or VizieR.

You can scroll up/down/left/right the contents of the measurement window just by clicking-and-dragging on it. You can also split the measurement window in a separate (resizeable) window, by clicking on the *out* arrow, at the top right corner of the measurement panel. It allows you to consult more easily long lines and long lists.

3.6. The tool bar

The tool bar provides a number of control tools to help visualize the data and analyze the images.

3.6.1. Properties

The Prop. button provides information about the properties of the selected plane (i.e. the one with a framed name): number of entries, for data planes; source and reference for image planes. The Properties window pops up, and it is possible to modify the configuration of the plane. In particular, you can change the colours, or the symbol used for displaying the objects.

The contents of the Properties window (Fig. 9) may change according to the plane type.

Note that it is now possible to adjust the symbol size according to a measurement of this object (section 5).

3.6.2. Draw

To draw lines or contours, use the Draw button. You have two modes:
- click-and-drag mode: draw a "hand-line"
- click-and-click mode: draw a "poly-line". You have to take the cursor out of the view to stop the polyline.

The drawing is memorized together with astronomical positions. It means that you can display it on another image not necessarily having the same resolution.

You can move a contour with the Select tool by clicking-and-dragging one of its control points. First select the contour and then click-and-drag one of the control points (little squares on the contour).

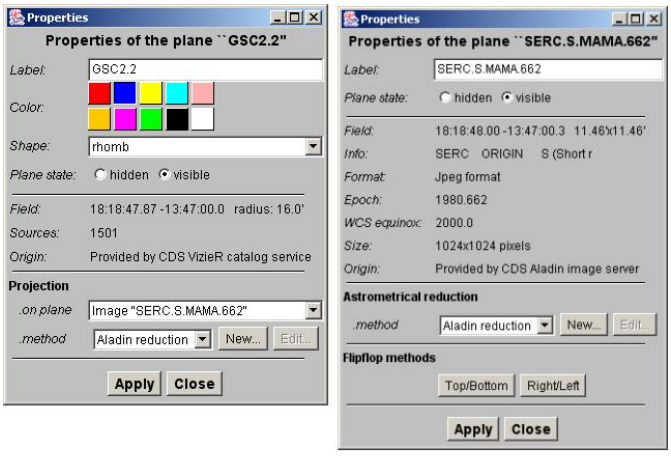

Fig. 9. Examples of Properties windows for a catalogue plane (left) and for an image plane (right). The colour and shape of the symbols from a data plane, displayed in the View window, can be modified by using the corresponding menus (left). Epoch and source of the image are given in the corresponding window (right). It is also possible to modify the projection of the data points, or the astrometric registration of the image.

3.6.3. Text, Tags

You can display text or mark objects with tags by pushing the related Text or Tag button.

3.6.4. Distance (Dist)

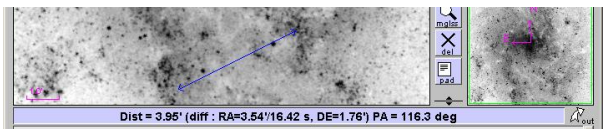

Fig. 10. Example of display obtained when using the Dist button.

The Dist button activates an "Overplot measurer" allowing you to draw vectors in the View window: the corresponding angular distance, differences in right ascension and declination, and position angle, appear in the Status panel below the View window (Fig. 10).

3.6.5. RGB colour composition

If you have load two or three images covering the same field, you can call the RGB colour composition by pushing the rgb button. A popup window will be displayed, allowing you to select which images will be assigned red, green and blue colours. Astrometric registration will be computed, using by default the highest resolution image as a reference.

You can also use the ALADIN previewer that can provide directly a color (RGB) image (Fig. 2).

3.6.6. Contour plots

The cont button can be used to draw isocontours, computed from the selected current image: these contours can be overlaid on other images as illustrated in fig. 11.

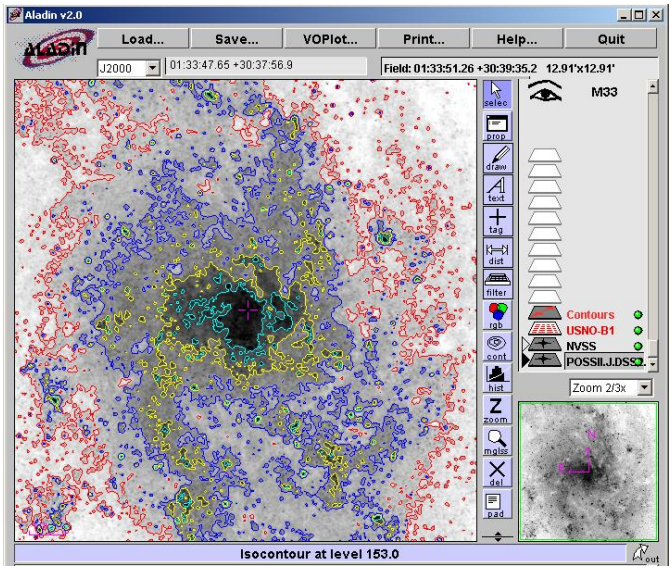

Fig. 11. Contour plot of M 33.

3.6.7. Colour map (Histog.)

You can modify the colour map of the image by clicking on the Histog. button (Fig. 12).

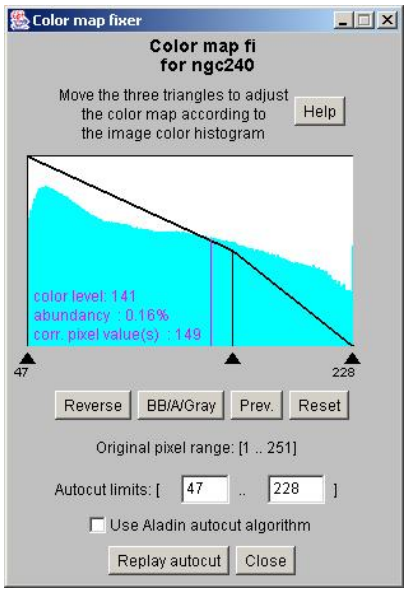

Fig. 12. Colour map panel. This window pops up when you push the Histog. button. It displays the histogram of pixel values of the current background image.

You can reverse colours (Reverse button), apply false colours (BB, A, Gray, similar to the corresponding features of DS9/saoimage), and adjust the pixels dynamics: move the left triangle towards the right to suppress the background noise; move the right triangle towards the left to enhance the objects; adjust the middle triangle according to the histogram to enhance the intermediate gray levels.

3.6.8. Zoom and image manipulation

The square window at the bottom right corner is the Zoom window; it displays a quick view of the current active image on the plane stack.

The green rectangle specifies the size of the current view seen in the large View window, according to the zoom scale and the total size of the ALADIN window. By dragging this rectangle, you can change the displayed area.

The zoom scale can be changed: i) with the little menu button, or ii) through the contextual popup menu (right button clicking in the View window), or iii) directly clicking in the View window if the Zoom button is activated.

The size of the View window depends on the total size of the Aladin Window. With the Applet version, click on the Detach button to separate Aladin from the browser, and by doing so, you will be able to maximize the window size.

To get a survey image adjacent to the current one, it is possible to use the Zoom tool with a factor less than 1. After that, use the Grab coord button of the image server form and select the center of the next image by clicking in the view margin.

3.6.9. Scale and coordinate grid

By invoking the contextual menu (popup menu: clicking the right button of the mouse; or CTRL click in the MacOS environment) in the view frame, you can display the scale and coordinate grid:

1. the scale: a small line gives the scale of the view;
2. the coordinate grid (for constant RA and DEC values);
3. the target: a large cross (+) indicates the object or position used as center in the initial query.

3.6.10. Cut-and-paste using the Pad window

Open the "Pad window" with the Pad button. This window stores information associated to the graphical planes (tags, distances, ...) and also data about astronomical objects selected with the mouse. It is available for cut-and-paste. For example, from the Pad, it is easy to export some data from Aladin to MS Excel.

3.7. Help

Opening the interactive HELP mode menu, move the mouse pointer on one of the interface components: a short help related to this component will be displayed.

Click elsewhere to quit the help mode.

4. More Detailed Features

4.1. Manipulating the plane stack

The plane stack allows you to control the current projection. Each plane keeps the result of a server query (image servers or data servers) or possibly some additional graphical objects, such as tags or drawings.

Click on a plane icon to ask to activate it according to the current projection; the icon is dark when the plane is active. Click on the eye icon to switch on/off projection of all possible overlay planes.

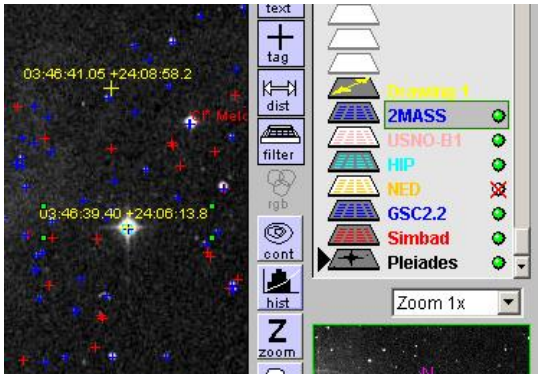

Fig. 13. A plane stack in a field featuring the Pleiades. Simbad, GSC2.2, USNO-B1, and 2MASS are the active data planes. NED has no entry in this (stellar) field (as can be seen from the status icon on the right). 2MASS is the selected plane (framed name), ready for Prop. or Cont. tools.

Click on a plane name to select it; then the tool buttons which are authorized for this kind of plane are activated. Maintain the Shift key to select several planes together.

Click and drag the plane icons to change the order of the display in the stack. The first image hides all planes underneath.

Delete the selected plane (the one with a framed name) by clicking on the Del button.

Each image or catalogue plane has its own projection parameters (target, radius, projection method). Only one of them is used at a time and other planes are projected accordingly. A little black triangle at the left of the plane logo shows which plane is used as the reference projection. Other planes which can also be used as reference projection are shown by an unfilled triangle. To change the current reference image of the view, click on the little triangle in front of the logo of the new image.

You may create a folder to sort your different planes (open a pop-up menu –right click with the mouse– from the plane window). Notice that filters (section 5) inside a folder are active only for images, below the filter, inside the folder.

4.2. Interacting with the VizieR database

The astronomical catalogues and tables are taken from the **VizieR** data base.

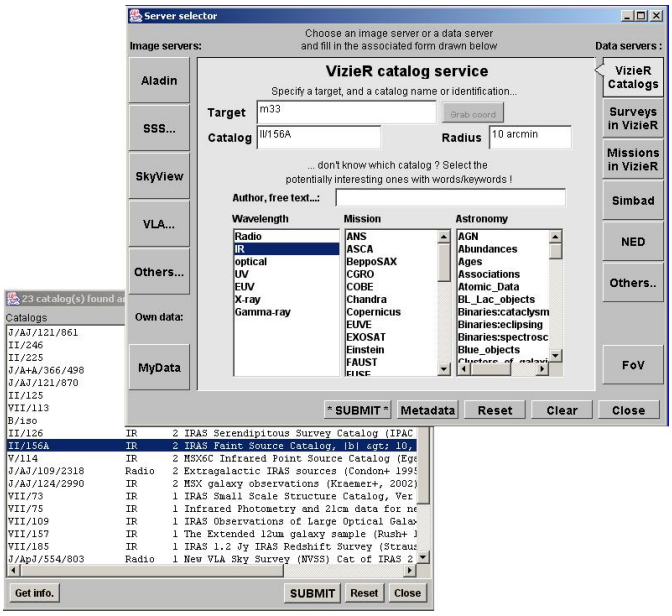

Fig. 14. VizieR query panel and a list of catalogues in the neighbourhood of the target, list provided by keywords.

The interface between ALADIN Java and VizieR presents different alternatives for the choice of catalogues. Once you have pushed the Load button, and selected the VizieR Catalogs button from the query window (Fig. 14):

- just type in the Target box the name object or its coordinates. Then the submit button provides a list of *all catalogues* which have a fair chance of having at least one source in the neighbourhood of the target (section 7.1). Catalogues in this list can then be selected by a mouse click on their title;
- type in the Target box the name object or its coordinates and type in the Catalogue box the acronym (e.g. `HD`) or CDS/ADC identification of catalogue (e.g. `III/135`); a comma-separated list of acronyms may be specified for several catalogues, e.g. `BD,CD,CPD`. *Browse the list of catalogues*[6] for a complete list of available catalogue acronyms or identifications;
- you may refine the list of catalogues by submitting additional constraints (Fig. 14):
  - by entering words in the Author, free text... box (e.g. author name), the list of catalogues is restricted to those having the specified word(s) in their description *and* having sources in the target neighbourhood.
  - by clicking on a selection of keywords, the list of catalogues is restricted to those matching *all* keywords *and* having sources in the target neighbourhood.

Other Vizier alternatives are offered from the Load panel (Fig. 14):

- select the Surveys button, the largest all-sky surveys available are listed, and can be selected by a mouse click on the corresponding line.

[6] http://cdsweb.u-strasbg.fr/cats/cats.htx

– the Missions button triggers display of the available observing logs (see 3.4.5).

Note that catalogues can be displayed in the View window without a background image, making possibly easier direct comparison of two catalogues or tables. In this case the projection is even more precise, as it is not adjusted to the pixel size.

4.3. Shortcut commands for loading data

As an alternative to using the Load button, shortcut commands can be submitted by clicking in the Position window, just below the menu (Fig. 15).

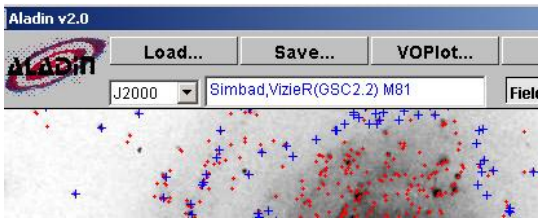

Fig. 15. Inserting a shortcut command (blue text on white background) in the Position window.

The shortcut commands follow this syntax (no space after the comma):

```
Get ServerName[,ServerName] Target
```

– **ServerName** can be *Simbad*, *NED*, *Aladin* or *VizieR*. For this last one, the catalogue specification is required (in parentheses). For Aladin, the survey, colour, or resolution can be specified as well between parentheses.
– **Target** can be a Simbad identifier or J2000 coordinates in sexagesimal syntax.

If the *Target* is omitted, Aladin takes into account the last specified target. If, for an initial query, there is only a *Target* and no *ServerName*, Aladin will load a default set (currently: DSS-I high resolution image, SIMBAD and GSC1.2).

You will find in the FAQ of the interactive HELP menu detailled possibilities for the syntax of shortcut commands.

Examples:

```
get Simbad,VizieR(GSC1.2) M 81
get Simbad 00 42 44.10 +41 16 08.8
get Aladin(DSS2),VizieR(USNO2) NGC 7436
get Aladin(DSS2),NED 18 19 -13 50
get Skyview(Rosat) M 1
get SSS(UKST) M 2
```

4.4. Coordinates manipulation

It is possible to visualize the location of a specific object or position in the current view, by simply clicking in the Position window, and typing the coordinates (or Simbad identifier). After pressing the 'RETURN' key, the current target (large +) will move at the corresponding position in the View window. Also if you click on the View vindow, the corresponding position will be automatically memorized in the Position window *and* in the note Pad window. Move the cursor in the Position window (or in the note Pad window) to retrieve this position and possibly cut-and-paste into another application.

When you move the cursor on the object data line in the measurement window, the position of this object is automatically given (in the current coordinate reference system) in the position window. You can easily display coordinates in another reference system using the nearby menu button: if you select, e.g., the Gal reference system and move the cursor on the object data line, you will see the position displayed in Galactic coordinates.

4.5. Using the SHIFT key for hidden features

In order to keep the Aladin interface as simple as possible, some features are hidden and require the usage of the SHIFT key. By this way, you can:

– *Remove several planes at a time*: specify several planes by clicking on their name, with the SHIFT key down. After that, press the Del button.
– *Remove all planes*: with the SHIFT key down, press the Del button.
– *Select several individual sources:* maintain the SHIFT key down and click on each source.
– *Select all sources of a plane*: specify the plane by clicking on its name and with the SHIFT key down press the Select button (possibly twice if there was previously some selected objects).
– *Open links in* new *browser windows*: click on the links with the SHIFT key down.
– *Force a catalogue projection*: in the plane stack, click on the left of the plane logo of a catalogue with the SHIFT key down (where there should have a little black triangle if it was an image plane).
– *Center the view*: click in the Zoom Frame (at the bottom-right of the screen) with the SHIFT key down to re-center the current view (only if the current image is larger than the view window).

4.6. URL command lines

It is possible to use a URL (Uniform Resource Locator) for sending input to the Aladin server. Here is an example:

```
http://aladin.u-strasbg.fr/AladinJava?script=get+
  Aladin,Simbad,VizieR(GSC2.2)+M50
```

Opening this URL from a browser triggers the display of a DSS-I view of the open cluster M 50, with SIMBAD and GSC2.2 overlays.

The detailed syntax of the input string following the question mark is the same as the syntax of the script language (see section 8.5).

An automatic generator of scripts and of URL helps you building more easily simple scripts and URL.

(see `http://aladin.u-strasbg.fr/java/nph-aladin.pl?frame=form`)

5. Filters in Aladin

5.1. Introduction

Filters are an elaborated feature in Aladin allowing one to customize the display of catalogue planes in Aladin.

Without filters, all entries from a catalogue plane are displayed with the same symbol and the same colour: the only parameters used, from the original catalogue, are the sky coordinates (RA and Dec) of each entry.

Thanks to the filters feature, it is possible to use the other parameters from the original catalogue (e.g., magnitudes, object types, velocities) in order to perform selections, and modify the symbol shape, color, size, or text, according to a parameter or a combination of parameters (using arithmetic operators: $+ - * / \hat{}$). The position of the symbols is always the sky position in the current field.

Figure 16 is an illustration of the possible use of a filter: bright sources have been circled ans labelled while magnified proper motions are plotted. This example can be built using and modifying predifined filters: MAGNITUDE-CUT, PROPER MOTIONS ...

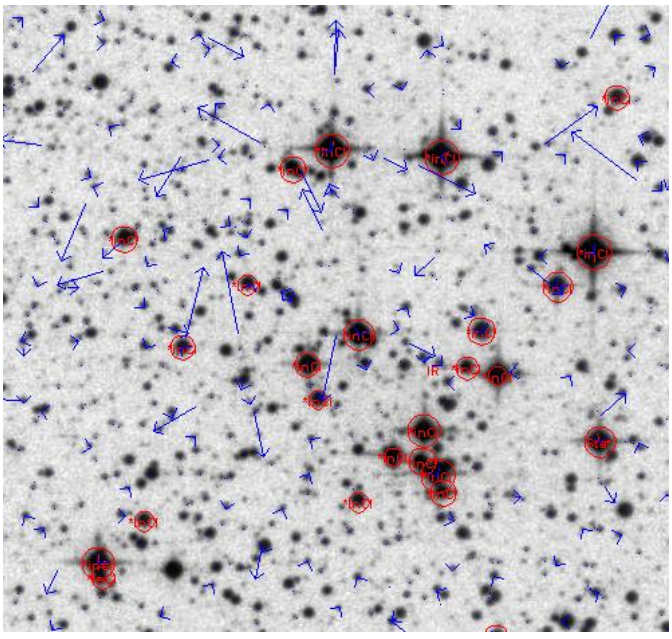

Fig. 16. Example of magnitude-circle and proper motions filter result

A list of predifined filters is proposed to cover some frequent situations and a pop-up menu helps you to build filter commands.

For instance the predefined MAGNITUDE CUT filter allows to draw the objects of a table having a magnitude smaller than 16. The magnitudes come from the first column for which the object has a photometric magnitude given by a UCD of the `$[PHOT*]` form.

For this example, the filter command appearing in the filter window has the form: `$[PHOT*]<16 {draw}` and can be easily modified. The UCD generic name `$[PHOT*]` may be replaced by any columns of loaded tables. The names of all the avalaible loaded columns during a Aladin session can be displayed by activitating a pop-up menu (right button mouse) in the command menu of the filter properties window. This allows to select and load a column name in the filter command window.

5.2. Syntax

A **Filter** consists of a list of **Constraints** and associated **Actions**.
Example of filter:

```
Constraint 1 { Action 1
               Action 2 }
Constraint 2 { Action 3 }
```

A **Constraint** is a set of **Conditions** combined by logical operators (AND, OR).
Example of constraint:

```
(Condition 1 && Condition 2) || Condition 3
```

A **Condition** applies to **Parameters** (or combination of parameters).
Examples of conditions:

```
Parameter >= Value
Parameter 1 - Parameter 2 = Value
```

Parameters are specified via the corresponding column names or UCDs (Unified Content Descriptors). *Examples of parameters:*
via UCD : `$[PHOT_JHN_V]`
via column name : `${Vmag}`

A **Value** can be a numeric value or a string value.

The global structure is quite similar to a *switch* structure in C which would have a *break* statement at the end of each *case*. It means that if a source verifies one constraint, the program will not run into following cases for this source.

Note: for a given constraint, the *list of actions* immediately follows the constraint and must be embraced by brackets `{}`. Each *action* has to be separated either by a **carriage return** or by a **semicolon**.

Figure 17 illustrates how a filter is applied to a set of sources. If a source verifies the first constraint, then each corresponding action is made and the next source is processed. If it doesn't, the program checks whether the source verifies the next constraint, and so on until there are no more blocks. If the source verifies no constraint at all, it is not drawn.

5.2.1. Invocating columns

In constraints and in the parameters of actions, UCDs and column names have to be pointed out in a specific way.

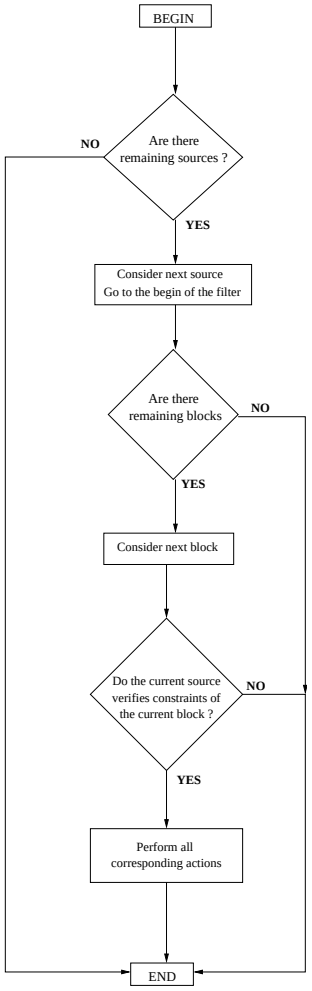

Fig. 17. How a filter processes a set of sources

The syntax is the following:

- Parameters always begin with `$` (dollar sign)
- UCDs are enclosed in square brackets `[]`. E.g: `$[POS_EQ_PMDEC]` or `$[PHOT_FLUX_RATIO]` are valid UCD names.
- Column names are enclosed in braces `{}`. `${LumRat}` or `${Imag}` are valid column names.

UCD names are case insensitive (in fact, they are automatically converted to uppercase) whereas column names are case sensitive.

The star `*` and the question mark `?` can be used as a wildcards in UCDs or column names. For instance, `$[PHOT*]` will point out the first column name being tagged by a UCD beginning with `PHOT`.

5.2.2. Comments

Comments may be inserted into filters. Each line beginning with "#" will be ignored. Example:

```
# This is a comment
```

5.2.3. Arithmetic expressions using UCDs/columns

The following operators are used to combine catalogue parameters: +, -, *, /, ^ . The parameters are specified via the corresponding UCDs or column names. They can be used to define constraints, but also in some action parameters. Example:

```
${Bjmag}-$[PHOT_JHN_V]>1 {
# Do some action
...
draw ellipse( $[EXTENSION_RAD],
 $[EXTENSION_RAD]*(1-$[PHYS_ECCENTRICITY]^2)^0.5,
 $[POS_POS-ANG] )
}
```

5.2.4. Conditions

A **Condition** is an arithmetic combination of **Parameters** followed by a **Comparison Operator** followed by a **Value**.

The table below summarizes the allowed comparison operators for a numeric value.

Operator	Meaning	Example
$=$	Equality	$= 1$
$!=$	Inequality	$!= 1$
$>=$	Greater or equals	$>= 12.0$
$>$	Strictly greater	> 12.0
$<=$	Less or equals	$<= 12.0$
$<$	Strictly less	< 12.0

The table below summarizes the allowed comparison operators for a string value.

Operator	Meaning (case sensitive)	Example
$=$	String equality	$= "Galaxy"$
$!=$	String inequality	$!= "UV"$

When performing a comparison on strings, only one parameter can be used, *i.e.* combination of parameters is not allowed in this case.

Note: there is a special condition called *undefined(ucd or column name)* which is true if the entered ucd/column name is not present for the current object. Example:

```
# Draw black symbols for sources with no Bjmag
undefined(${Bjmag}) {
    draw black
}
# Process sources with column Bjmag
...
```

You can also use the following functions:

- *abs* : absolute value
- *cos* : cosinus
- *deg2rad* : convert degrees to radians
- *ln* : natural logarithm (base e)
- *log* : base 10 logarithm
- *rad2deg* : convert radians to degrees
- *sin* : sinus
- *tan* : tangent

E.g:

```
log(abs(${Fi})/${Fx})>44 {draw}
```

5.2.5. Constraints

The logical operators used to combine conditions are **&&** – logical AND – and || – logical OR. Example:

```
($[PHOT_PHG_B]<16 && $[PHOT_PHG_R]<15)
     || $[CLASS_OBJECT]="Star" {
    draw
}
```

A constraint can also be empty. In such a case, the corresponding actions apply to all sources. Example:

```
# No constraint, the action block
#     is applied to all sources
{draw blue}
```

5.2.6. Actions

Actions change the appearance of symbols in the related catalogue plane. Available actions are:

- *hide*: does not display the entry. This action is useful to hide a part of the plane sources.
- *draw ...*: when used without parameters, this action draws the source as usual using the default shape and color of the plane. The action has two optional parameters for customizing both *shape* and *color* of the symbols. Example: `draw green square`
- You can also draw a text *string*, which may be either a constant string or the content of a column. The color can also be specified. Example: `draw $[CLASS_OBJECT] blue`

5.2.7. Defining a color

You can specify the color you want to assign to either a symbol or a text string. The color function is optional and can be placed either after the *draw* keyword, or after the optional shape function. Example:

```
{
# Color function after the "draw" keyword
draw blue square
# Color function after the shape function
draw circle(-$[PHOT_PHG_B]) #00ff00
}
```

There are different ways to define a color:

- #rrggbb, where rr, gg, bb are respectively the values of red, green and blue components of the color in hexadecimal. Example: `{draw #44dd99}`
- predefined color names: allowed values are black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red white, yellow. Example: `{draw red}`
- *rgb* function allows to define a color whose components depend on some column values. For each component, the max and the min values are computed, so that each value is normalized in order to be in the range 0-255. Example: `{draw rgb($[PHOT_PHG_B],0,-$[PHOT_PHG_B])}`
- *rainbow* function has a single parameter in the range 0-1. This filter aims at going through the whole visible spectrum (red to violet, 0 to 1). If the parameter is variable (for instance a column name), the values are normalized between 0 and 1. This can be used to visualize a color index. Example: `{draw rainbow($[PHOT_JHN_U-B])}`

5.2.8. Symbol shape

Symbol shape can be customized, through the call of an optional parameter, placed either just after the *draw* keyword, or after the color function.

There are shapes without parameter: they are the same as those in the properties of a catalogue plane: square, rhomb, cross, plus, dot, microdot. Example:

```
# Filter drawing different shapes
# according to the object class

# Draw a plus for Star objects
$[CLASS_OBJECT]="Star" {draw plus}
# Draw a rhomb for Radio objects
$[CLASS_OBJECT]="Radio" {draw rhomb}
# Draw a dot for other objects
{draw dot}
```

In addition, there are some shape functions requiring parameters:

* the *circle* function is used to draw a circle whose radius depends on a parameter which can be any column or combination of columns. The typical use of this function is drawing circles according to magnitude values. There are 2 optional parameters to define the minimum and the maximum *radius* size **in pixels** (otherwise, these values are set by default). Parameter values are normalized to fit into the range. Example:

  ```
  # Draw circles according to Johnson B mag.
  {draw circle(-$[PHOT_JHN_B])}

  # Draw circles according to Johnson B mag.
  # Set min value to 3, max value to 40
  {draw circle(-$[PHOT_JHN_B],3,40)}
  ```

* the *fillcircle* function is similar to the *circle* one. It draws a filled circle instead of an empty one.
* the *ellipse* function is typically used to draw dimension ellipses or error ellipses. It takes 3 parameters : the semi-major axis, the semi-minor axis. If the unit of one parameter is missing or not understood, semi-major and semi-minor axis are assumed to be in arcsec, and the position angle is assumed to be in degrees. If the position angle can't be retrieved or is empty, its value is assumed to be zero. Example:

  ```
  # Drawing dimension ellipses for Simbad
  #   and GSC2 sources
  {
  # Ellipses for Simbad
  draw ellipse(0.5*${DimMa},
            0.5*${DimMi}, ${DimPA})
  ```

```
# Ellipses for GSC2
draw ellipse( $[EXTENSION_RAD],
 $[EXTENSION_RAD]*(1-
$[PHYS_ECCENTRICITY]^2)^0.5,
 $[POS_POS-ANG] )
}
```

* the *pm* function was designed to allow the visualization of proper motion of stars. It takes 2 parameters which are the proper motion in right ascension and the proper motion in declination, and draws an array corresponding to this proper motion. If units are missing or could not be understood, both parameters are assumed to be in mas/yr. By default, a proper motion of 1mas/yr is displayed by an array of 1arcsec.
E.g:

```
# Draws proper motions
# 1mas/yr will correspond
#          to an array of 5 arcsec
{draw pm(5*$[POS_EQ_PMRA],5*$[POS_EQ_PMDEC])}
```

5.3. Usage in Aladin

5.3.1. Creating a filter

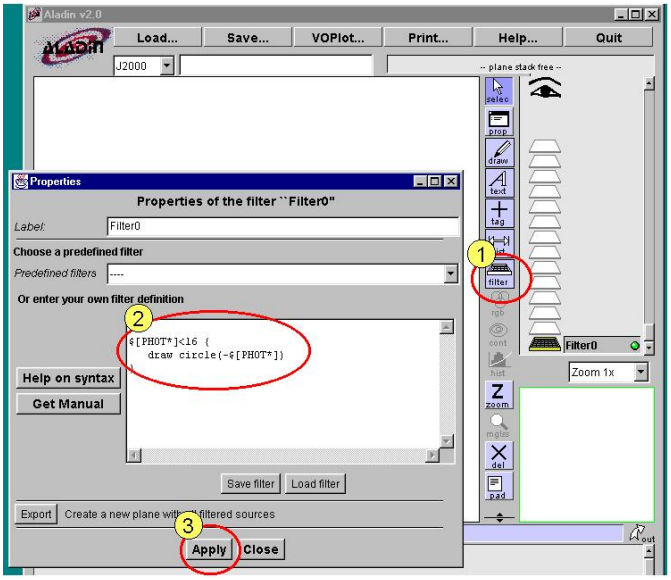

Fig. 18. Creating a filter in 3 steps

Figure 18 describes how to create a filter. First, click on the Filter button in the tool bar. This will create in the stack a new plane dedicated to a filter. The *Properties* window will pop up, in order to let you enter your definition. Type in your filter definition or choose a predifined filter, and press the *Apply* button.

In the properties window of a filter, predefined filters are intended to help you understand the syntax of a filter.

5.3.2. Modifying a filter

To modify a filter, open the *Properties* window of the filter. Then, modify the definition and press *Apply* to confirm the changes. If the filter is active, the result will be updated. If it is not, the new definition will be taken into account as soon as the filter becomes activated.

5.3.3. Syntax errors

When creating or modifying a filter, one may enter a definition which is syntactically incorrect. If it happened, the filter is deactivated, and a message pops up announcing the detected error. At the same time, the status disc next to the filter name in the stack, turns to red.

If you try to activate a filter whose definition is incorrect, you will see a message asking you to correct the error.

5.3.4. Activate/Deactivate a filter

Filters are like other planes: you can easily activate or deactivate them, just by clicking on the logo.

5.3.5. Scope of a filter

A filter applies to all *active* catalogue planes located below it in the stack. If the filter is in a folder, it applies to all active catalogues located below it within the same folder, including catalogues being in subfolders.

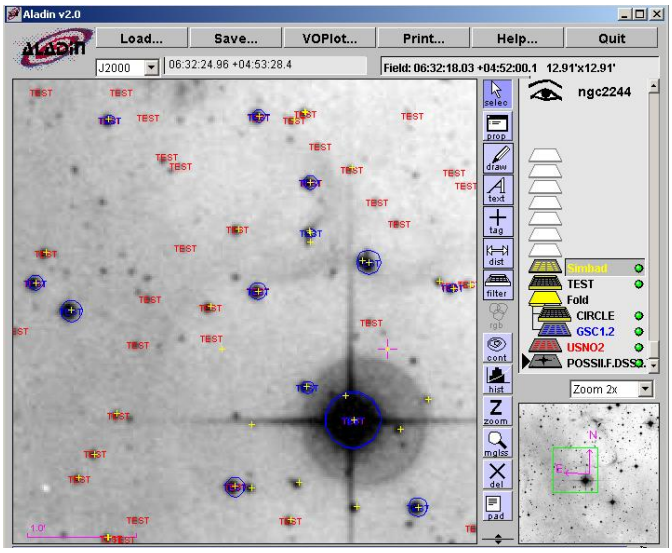

Fig. 19. Scope of filters (see text)

Figure 19 explains this principle. Two filters, *TEST* and *CIRCLE*, have been created. *TEST* just prints the text string **TEST**, and *CIRCLE* draws a circle according to the magnitude of the object.

As you can see, the filter *TEST* applies to planes GSC1.2 and USNO2 which are below it, but does not apply to plane Simbad. *CIRCLE* applies to GSC1,2, which is in the same

folder, but does not apply to USNO2 even though it is located below *CIRCLE*.

Whenever a filter or a catalogue plane has moved, whenever a new catalogue plane is created, filters results are automatically reprocessed and updated.

5.3.6. Applying multiple filters

You can apply several filters simultaneously. Each filter has its own scope, and each filter performs its actions apart from each other.

5.3.7. Creating a filter in script mode

Filters can be used in script mode, via in-line commands. The syntax is almost identical. The only difference is that the command starts with `filter filtername {` and ends with a closing bracket `}`. The filter definition can be on multiple lines: the prompt shows `Aladin - Filter def.` until the final `}` is entered.
Example:
Aladin> `filter circle {`
Aladin - Filter def.> `{draw circle(-$[PHOT*])}`
Aladin - Filter def.> `}`

5.3.8. Miscellaneous

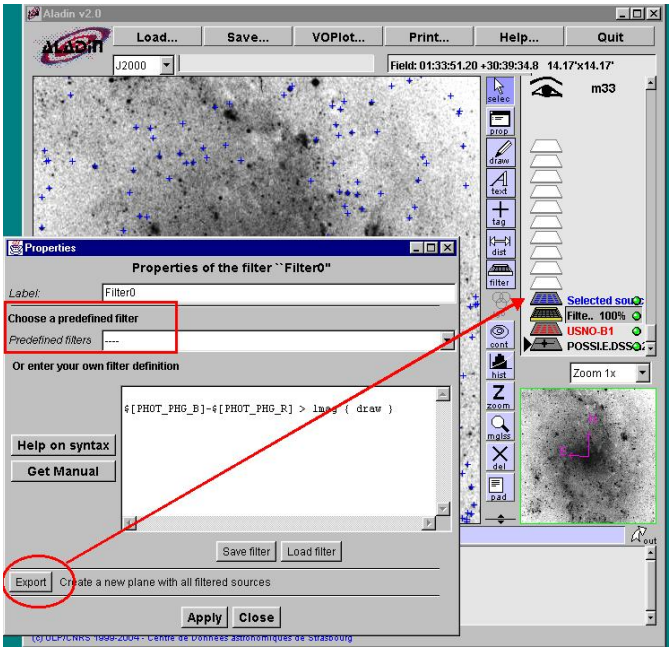

Fig. 20. Export feature

Figure 20 shows buttons contained in the *Properties* window of a filter.

– *Export*: creates a new catalogue plane containing all filtered sources. This can be useful to save the result of a filtering process.

The names of all the avalaible loaded columns during a Aladin session can be displayed by activitating a pop-up menu (right button mouse) in the command menu of the filter properties window. This allows to select and load a column name in the filter command window.

6. The image database

6.1. Database summary

The ALADIN image collection consists of:

– The whole sky image database from the first Digitized Sky Survey (DSS-I) digitized from photographic plates and distributed by the Space Telescope Science Institute (STScI) as a set of slightly compressed FITS images (with a resolution of 1.8 arcsec.);
– DSS-II: full sky R and I-band, and Palomar B-band surveys scanned at STScI with a pixel size of 1 arcsec.;
– Images of crowded fields (Galactic Plane, Magellanic Clouds) at the full resolution of $0.67''$, scanned at the *Centre d'Analyse des Images* (MAMA machine), Observatoire de Paris; the ESO-R survey also digitized with the MAMA will be soon included in the data base;
– Global plate views ($5\,\mathrm{deg} \times 5\,\mathrm{deg}$ or $6\,\mathrm{deg} \times 6\,\mathrm{deg}$ according to the survey) are also available for all the plates contributing to the image atlas: these are built at CDS by averaging blocks of pixels from the original scans;
– Low resolution images: Aladin image server also provides DSS-I images in low resolution (generally 1.5 x 1.5 degrees);
– The 2MASS J, H, and K_S images. The sky survey have been released by the *2MASS Project*[7] (Incremental Release 2) and is progressively included in the Aladin image collection.

Access is also provided to other image servers: currently SkyView, SuperCosmos, and several mirror copies of the DSS image service.

Other image sets are available through the Missions in Vizier feature (section 3.4.5): currently HST or FIRST images can be retrieved and displayed.

Finally, user-provided images, in FITS format, having suitable World Coordinate System information in the header can also be used through Aladin; this functionality is available browsing your local directories for the Java stand-alone version (section 8.2). It is also available for the Java applet version for personnal files accessible through a URL (section 3.4.7).

6.2. DSS Images

The database currently includes the first and second Digitized Sky Survey (DSS-I and DSS-II) produced by the Space

[7] http://www.ipac.caltech.edu/2mass/

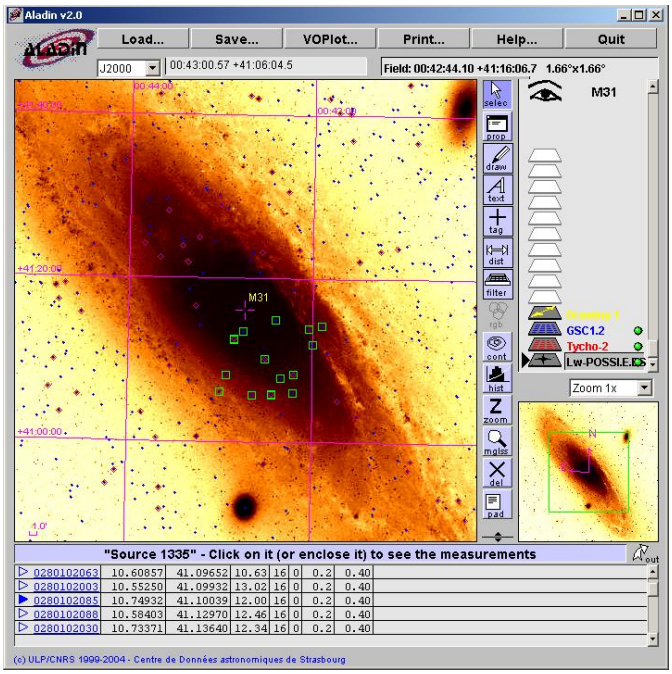

Fig. 21. Low resolution image of M31: the field size is $1.6°$ (image from DSS-I). Highlighted symbols in the display (green square) show selected objects, for which the data (from GSC and Tycho-2) are displayed in the Measurement window. A filled blue triangle in this window signals the GSC entry closest to the cursor.

Telescope Institute for the needs of the Hubble Space Telescope.

To create the DSS-I images, the STScI team scanned the first epoch (1950/1955) Palomar E Red and United Kingdom Schmidt J Blue plates (1980–), including the SERC J Equatorial Extension and some short V-band plates at low galactic latitude, with a pixel size of 1.7 arcsec. ($25\mu m$). Images are slightly compressed (factor of 10).

DSS-II images in the R-band come from Palomar POSS-II F and the UK Schmidt SES, AAO-R, and SERC-ER, scanned with a 1 arcsec. ($15\mu m$) sampling interval. DSS-II images in the B-band come from POSS-II J. DSS-II images in the I-band come from Palomar POSS-II N and SERC-I. The blue plates POSS-I O are progressively included.

6.3. MAMA/CAI Images

High resolution digitization of POSS-II, SERC-J, SERC-ER, SERC-SR, SERC-I, or ESO-R plates featuring crowded regions of the sky (Galactic Plane and Magellanic Clouds) have been provided by the MAMA facility at the *Centre d'Analyse des Images* (CAI), Observatoire de Paris. Sampling is $0.67''$ per pixel ($10\mu m$).

These images are better suited to visual inspection of very faint objects.

6.4. 2MASS near-infrared survey

2MASS J, H, and K_S images are the original, slightly compressed images from the 2MASS survey. They are currently *not* recomputed: this means that the retrieved image field includes the requested position, but is not centered on this position, as this is done for the high-resolution DSS images.

6.5. The image server

DSS images are generally available in the following resolutions:

- the Full resolution (about 10x10 arcmin) corresponding to the original digitizations;
- the Low resolution (about 1.5x1.5 degrees) providing an intermediate view consistent with the field of view of major telescopes (DSS-I images only);
- the Global Plate view (about 5x5 degrees) reproducing the photographic Schmidt plates.

The image server for ALADIN is able to deal with various survey data, in heterogeneous formats (uncompressed FITS, compressed JPEG, etc.).

High resolution images are currently stored as subimages of 500×500 pixels (DSS-I), 768×768 pixels (DSS-II), or 1024×1024 pixels (MAMA).

6.6. Image astrometry

Astrometric information comes from the FITS header of the DSS image, and is generally accurate to the arcsecond.

However, in some cases, plate astrometry can be off by up to $4''$ preventing an absolute astrometric usage. The main reasons are:

1. the pixel is $1.7''$ wide for the STScI images and $0.67''$ wide for the MAMA images ;
2. the error on the position given by the DSS-I calibrations (STScI images) can be of the order of $4''$ on the plate edges.

The accuracy of the positions of one given image can easily be checked by superimposing reference astrometric catalogues such as the *Tycho-2 Catalogue*[8] or the *US Naval Observatory Catalogue of Astrometric Standards (USNO-A2.0)*[9].

You can visualize the FITS header only for the images downloaded in FITS format. Select the plane of this image in the plane stack by clicking on its name and press the Prop. button. You will find a button to visualize its FITS header.

6.7. Image compression

For the ALADIN Java interface and for the ALADIN previewer, the current choice is to provide the user with a slightly compressed image, in JPEG 8-bit format, constructed from the original uncompressed FITS images. JPEG is a general purpose standard which is supported by all current Internet browsers.

[8] http://vizier.u-strasbg.fr/viz-bin/Cat?Tycho-2
[9] http://vizier.u-strasbg.fr/viz-bin/Cat?USNO

The size of such an image does not exceed 30 kBytes, and thus the corresponding network load is very small.

You may load the original digitized FITS images of POSS surveys, mostly to examine the faintest objects or faint and extented objects. This is done after pushing the submit button (Fig.4) and selecting the FITS format into the Info Frame window.

7. The catalogue and survey server

7.1. VizieR

The ability to access all VIZIER multi wavelength catalogues and tables directly from ALADIN is a unique feature which makes it an extremely powerful tool for any cross-identification or classification work.

The request of a catalogue around a target relies on a special feature – the genie of the lamp: this is the ability to decide which catalogues, among the database of (currently) about 4000 catalogues or tables, contain data records for astronomical objects lying in the selected sky area. In order to do that, an index map of VIZIER catalogues is produced (and kept up-to-date), on the basis of about ten pixels per square degree: for each such 'pixel' the index gives the list of all catalogues and tables which have entries in the field.

When a user requests a catalogue around a target, this index is queried and the list of useful catalogues is returned. It is possible, at this stage, either to list all catalogues, or to produce a subset selected on the basis of keywords, as explained in section 4.2.

Note that, as the index "pixels" generally match an area larger than the current sky field, there is simply a good chance, but not 100%, to actually obtain entries in the field when querying one of the selected tables.

7.2. Archive previews

In order to access image from archives, you should first query the VizieR service for the mission log of the archive (for example, the HST log or the FIRST log) around a target (see section 3.4.5).

Secondly, you select the resulting positions in the View window and look at the corresponding data in the Measurement window. If you find a button Image in the data line, click on it to retrieve the corresponding archive image (Fig. 7). A new plane will be created automatically in the stack.

8. Specific features of the Stand-alone version

ALADIN Standalone is a program that you can download in order to run Aladin Java as a standalone application on your local machine.

The main advantage is to overcome Java security restrictions: you can load and save image/data on your disk, print fields, etc. If you choose this solution, you should first download the *Aladin Java package*[10].

[10] http://aladin.u-strasbg.fr/AladinJava?frame=downloading

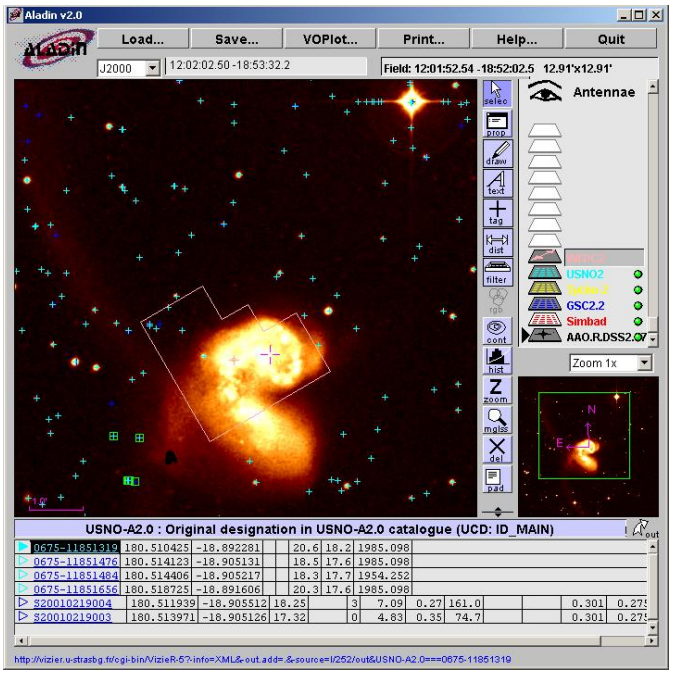

Fig. 22. A view of the Aladin Standalone version featuring the Antennae galaxies. Objects present in Tycho-2, GSC, and USNO A2.0 are flagged with different symbols, and a contour of the field of view of the HST WFPC2 camera has been displayed using the FoV function. Field size is 14 arcmin. The Save and Print buttons in the top menu are typical of the Standalone version.

It is possible to Save the work session (in a proprietary Aladin Java AJ format) for future re-use, or to Print a postscript version of the main View (Fig. 23).

8.1. Starting the Stand-alone version

ALADIN Standalone version requires a *Java Virtual Machine*. To be sure that you have a sufficiently recent one, download the Aladin Java package that includes a *Java VM*.

8.1.1. The Java Virtual Machine

The Java binary code generated by a Java compiler is independent from any hardware and operating system. To be executed, it needs a *virtual machine*, i.e. a program analyzing the binary code and executing the instructions. This virtual machine is dependent on hardware and operating system. Two types of Virtual Machines exist: those included inside current Web Browsers, and those running as an independent program.

8.1.2. The AladinJava Stand-alone package

The AladinJava Standalone package is packaged by InstallAnywhere (by ZeroG, http://www.zerog.com). The package includes the Aladin code and an embedded Java Virtual Machine.

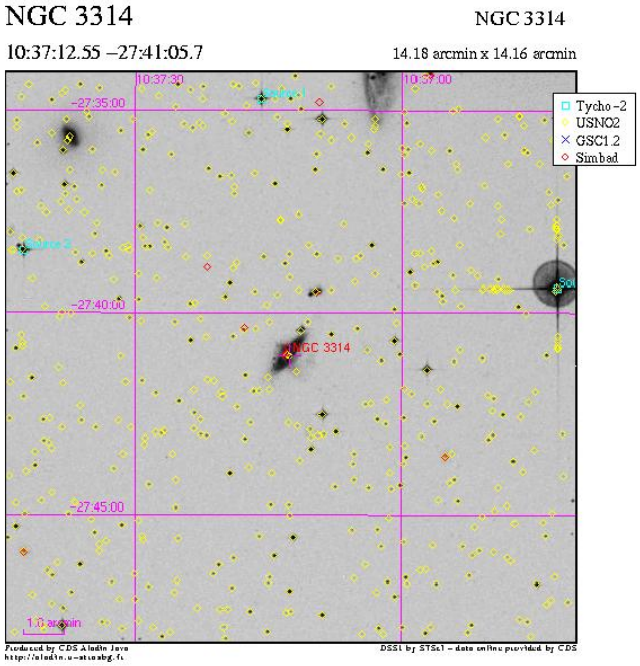

Fig. 23. Postcript printout of a DSS-I field featuring NGC 3314, produced from the Standalone version.

The installation does not require any special privilege but depends on your operating system (see the Aladin standalone installation web page).

Executing the Aladin Standalone code will also depend on your operating system: for instance an "icon menu" under Mac and Windows, "Aladin" command under Unix, etc... The standalone version can also be started with a java command line: "java -jar Aladin.jar"

Notice that InstallAnywhere provides a script to uninstall the codes.

8.2. Loading personal files with Aladin Standalone

The stand-alone version allows you to load personal images or tables, stored on your local computer hard disk.

8.2.1. Local images

Images should be in FITS format, with *WCS fields*[11] in the header (see e.g. Greisen & Calabretta 1995).

Note that you have to increase your JVM memory if you need to load large files. To do that you can play with two parameters of the java command:

`-ms<number>`: set the initial Java heap size
`-mx<number>`: set the maximum Java heap size

For example, a command like: `java -ms800m -mx800m -jar Aladin.jar` might allow you to manipulate the large CFH12K images (400MB).

[11] http://www.cv.nrao.edu/fits/documents/wcs/wcs.html

8.2.2. Calibrating Images

Images without astrometric calibration can be loaded. In order to visualize and overlay catalogues on them you will need to build an astrometric calibration.

To calibrate or to redo the calibration of any images, astrometric calibration tools are available in Aladin. The astrometric reduction menu is activated from the properties menu associated to your image plane. You will also need astrometric references from another calibrated images of the same area or from a corresponding astrometric catalogue. Just clicking a few common stars to your uncalibrated image and to your reference frame image or catalogue can be sufficient to obtain a rough astrometric calibration (examples for image calibrations are given at the address: http://aladin.u-strasbg.fr/tutorials).

8.2.3. Local Tables

Tables should be in one of the following formats:

– **Tab-Separated-Value** (TSV): one record per line, each field separated by a TAB. By default, the first column is assumed to be Right Ascension, and the second column Declination (both in J2000 equatorial system). Values are to be given in decimal degrees, or in sexagesimal hours and degrees. Heading line can be present, as shown below:

```
RAJ2000 DEJ2000 GSC number      Pmag
------- ------- ----------      -----
185.701 15.822  0144501972      15.55
185.766 15.795  0144502507      13.92
185.704 15.844  0144501918      15.06
185.710 15.849  0144502383      14.78
```

– **XML**: The **VOTable XML** format is developed by the International Virtual Observatory in Astronomy (IVOA). It allows to describe astronomical catalogues with **XML**. Aladin uses it to retrieve data from *VizieR*, *Simbad*, *NED*, or to load local data files. The **VOTable** format is detailed at the following address: http://vizier.u-strasbg.fr/doc/VOTable.
Aladin also supports **AstroRes**, the predecessor of **VOTable**. **Astrores** format is detailed at the following address: http://vizier.u-strasbg.fr/doc/astrores.htx

8.3. Saving and Printing

It is possible to Print a postscript version of the main View (Fig. 23). Different options are offered to Save data from an Aladin session:

– The work session can be saved (in a proprietary Aladin Java AJ format) for future re-use
– Catalogues or images loaded in the plane stack can be saved respectively in ascii or fits formats.
– A file view with the current overlays can be saved in bmp format.
– An HTML page with links to the original images loaded in the plane stack can also be created.

8.4. Defining additional servers

The user of the Standalone version can define and query new data or image servers, besides Simbad, Aladin or VizieR.

To do that, you have to append the new server definitions in the configuration file `AlaGlu.dic` present in the Standalone package, and restart Aladin.

The servers have to be accessible by a simple URL with a HTTP GET method. The syntax required for these server definitions follows the GLU recommendations. Adapt this short example to your own needs:

```
%ActionName          Foo
%Description         My own server definition
%Aladin.Label        MyServer
%Aladin.Menu         Others...
%Aladin.LabelPlane   MyServer $1/$2
%DistribDomain       ALADIN
%Owner               CDS'Aladin
%Url                 http://xxx/yyy?ra=$1&dec=$2
                          &radius=$3&color=$4...
%Param.Description   $1=Right Ascension
%Param.Description   $2=Declination
%Param.Description   $3=Radius
%Param.Description   $4=Color
%Param.DataType      $1=Target(RA)
%Param.DataType      $2=Target(DEC)
%Param.DataType      $3=Field(Radius)
%Param.Value         $3=14.1 arcmin
%Param.Value         $4=Red
%Param.Value         $4=Blue
%Param.Value         $4=Infrared
%ResultDataType      Mime(image/fits)
```

- **ActionName**: Unique identifier
- **Description**: Server description
- **Aladin.Label**: Button or menu label
- **Aladin.Menu**: label or upper-level button (in case of a submenu)
- **Aladin.LabelPlane**: Template of the plane label. `$n` is to be replaced by the corresponding parameters
- **DistribDomain**: keep this line "as is"
- **Owner**: keep this line "as is"
- **Url**: Server url template with $n variables specifying the fields which have to be filled up by the users
- **Param.Description $n=**: Description of the field number *n*
- **Param.DataType $n=**: Data type of the corresponding parameter. The available types are: Target(COO—RA—DE—SIMBAD—NED) and Field(RA—DE—SQR—RADIUS)
- **Param.Value $n=**: Default value, or list of possible values, for the corresponding parameter
- **ResultDataType Mime(*xxx*)**: To specify the data type provided by the server. For images: image/fits (fits file), image/gfits (gzipped), image/hfits (h-compressed), and for data: text/plain, text/tsv or text/csv (tab separated value syntax) and text/xml for data in XML/strores syntax (see section 8.2.3).

The servers that you have defined will be accessible by the buttons at the left side (images) or right side (data) of the server selector window, in your local Aladin standalone version.

8.5. Script language

In order to use Aladin Java in script mode, Aladin can be controlled via in-line commands. These commands have to be submitted to the `Aladin>` prompt which appears just after starting the application (only with the stand-alone version). Script commands can also be entered in the Position window (section 4.3). Additionally, the window interface can be removed by launching Aladin with the `-script` parameter.

The in-line commands are:

- `get servers [target] [radiusUnit]`: execute a shortcut command (see also section 4.3) to call image and data servers
  (ex: '`get aladin(DSS2),VizieR(GSC) M99`')
- `load` *`file`*: create a new plane with the file (image or data)
- `sync`: wait until all planes are ready
- `save` *`file`*: save the current view (FITS format; after `sync`)
- `export` *`nn file`*: export the plane number *nn* (BMP formats; after `sync`)
- `backup` *`file`*: generate a backup (proprietary AJ format) of the Aladin plane stack (after `sync`)
- `zoom` *`fct`*: change zoom factor on the current view (`1/2x`, `4x`, ...)
- `reverse [on/off]`: (un)reverse the current view
- `flipflop [V|H]`: vertical (resp. horizontal) pixel image inversion
- `cm` {`gray|BB|A`}: select the colour map
- `RGB [`*`n1 n2 [n3`*`|diff]]`: create a RGB image from plane number *n1*, *n2* [,*n3*] a "*" after a plane number specifies the ref plane
- `contour [`*`nn`* `[[no]smooth] [zoom]]`: draw *nn* isocontours on the selected plane (default: 4 levels) (zoom: draw the contours only on the zoom area).
- `filter [`*`name`*`]` {*`filter-content`*}: create a filter, the *filter-content* must follow the filter instruction rules (see section 5.3.7)
  (ex: `$[PHOT*]<16 {draw}`)
- `filter [`*`name`*`] [on/off]`: switch on/off the specified filter
- `md [`*`name`*`]`: create a new folder
- `mv` *`nn1|name1 [nn2|name2...] mm99|name99`*: move planes *nn*|*name* below plane *mm99*|*name99* (or into if *mm99*|*name99* is a folder)
- `rm [`*`nn|name`*`]`: remove a plane (the selected plane or the plane *nn*|*name*) or a folder and its content
- `select` *`nn|name`*: select the plane *nn*|*name* (number 1 is the bottom plane)
- `hide [`*`nn|name`*`]`: hide a plane (the current selected plane or the plane number *nn*) or the plane named *name*
- `show [`*`nn|name`*`]`: show a plane (the selected plane or the plane number *nn*|*name*) number *nn*)
- `ref` *`nn|name`*: set the plane *nn*|*name* as the reference plane
- `reset`: reset the Aladin plane stack
- `grid [on/off]`: (un)activate the coordinate grid
- `reticle [on/off]`: (un)activate the reticle

- `info` *`msg`*: print a message in the status window
- `scale [on/off]`: (un)activate the scale line
- `status`: display stack plane status
- `help`: display this help
- `pause [`*`nn`*`]`: wait *nn* seconds (default: one)
- `sync`: wait until all planes are ready
- `timeout` *`nn`*`|off`: set *nn* minutes as timeout (default: 15)
- `mem`: display the current memory size
- `quit`: stop Aladin.

Script example

Suppose that the file foo contains the following lines:

```
grid
get aladin(dss2),vizier(usno2) m99
sync
zoom 2x
cm BB
save m99.bmp
quit
```

By redirecting the Aladin standard input as below:

```
java -jar Aladin.jar -script < foo
```

Aladin produces transparently an image file called m99.bmp with a DSS2 image and the USNO2 catalogue overlay.

Launching Aladin with predefined images or data

The script commands can also be used in the URL (see section 4.6) to directly launch a predefined set of images and data. For example, the following URL will launch Aladin, for a field around NGC 1097, with a DSS2 image, SIMBAD, NED, and USN02 data, on reverse mode, and a coordinate grid:

```
http://aladin.u-strasbg.fr/AladinJava?script=get+
  Aladin(DSS2),Simbad,NED,VizieR(usno2)+NGC1097;
  grid+on;sync;reverse
```

Local images can be called by using the following script command: `get MyData(`*`YourURL`*`)`.

9. User feedback

The users play an important role by giving feedback on the desired features and user-friendliness of the interfaces, by mail to question@simbad.u-strasbg.fr.

New developments are currently considered as additional modules which will be incorporated to the general release only when needed, possibly as optional downloads, in order to keep the default version simple and efficient.

10. Acknowledgements

CDS acknowledges the support of INSU-CNRS, the Centre National d'Etudes Spatiales (CNES), and Université Louis Pasteur.

Collaboration with STScI, and especially with the late Barry Lasker, and with Brian McLean, is gratefully acknowledged. We thank Jean Guibert and René Chesnel from CAI/MAMA for their continuous support to the project. We thank Roc Cutri (IPAC) for his kind help in setting up the access to 2MASS images and data. Implementation of external links has been made possible by the collaboration of the service producers.

The Digitized Sky Survey was produced at the Space Telescope Science Institute under U.S. Government grant NAG W-2166. The images of these surveys are based on photographic data obtained using the Oschin Schmidt Telescope on Palomar Mountain and the UK Schmidt Telescope.

Java is a registered trademark of Sun Microsystems.

References

Bonnarel, F., Fernique, P., Bienaymé, O., et al., 2000, A&AS 143, 33 (Aladin)

Genova, F., Egret, D., Bienaymé, O., et al., 2000, A&AS, 143, 1 (CDS)

Greisen, E.W., Calabretta, M., 1995, in *Astronomical Data Analysis Software and Systems IV*, ASP Conf. Ser. 77, p. 233 (FITS WCS)

Guibert, J. 1992, in *Digitized Optical Sky Surveys*, H.T. MacGillivray & E.B. Thomson (Eds.), Kluwer Academic Publ., p. 103 (MAMA)

Ochsenbein, F., Bauer, P., Genova, F., 2000, 143, 23 (VizieR)

Wenger, M., Ochsenbein, F., Egret, D., et al., 2000, A&AS, 143, 9 (Simbad)

Appendices

Appendix A: Backus Naur Form of filter syntax

This is an attempt to describe the syntax of a filter in the extended Backus-Naur-Form. It describes in a shorter way what has been explained in section 5.2.

```
<filter> ::= <constraints block>+

<constraints block> ::= <constraint> "{" (<action><action separator>)* "}"

<action separator> ::= Carriage Return | ";"

<constraint> ::= ( <simple condition> [ <logical operator> <constraint> ] )
                        | "undefined(" (<UCD>|<column>) ")"

<condition> ::= <expression> <comparison operator> <value>

<expression> ::= ( ["+"|"-"] (<UCD>|<column>|<numeric>)
                 [ ( ("+"|"-"|"*"|"/") <expression>  | "^" <numeric> ) ] )
                 | <function>"("<expression>")"

<function> ::= abs | cos | deg2rad | ln | log | rad2deg | sin | tan

<value> ::= <string> | ( <numeric> [<unit>] )

<logical operator> ::= "&&" | "||"

<comparison operator> ::=  "==" | "!=" | ">" | ">=" | "<" | "<="

<action> ::=
"draw" [ <color function> ] [ (<shape function>|<UCD>|<column>|<string>) ]
| "hide"

<UCD> ::= "$[" UCD name "]"

<column> ::= "${" column name "}"

<shape function> ::= (("circle"|"fillcircle")"("<expression> ","<numeric>","<numeric>]")")
                   | "ellipse(" <expression> "," <expression> "," <expression> ")"
                   | "pm(" <expression> "," <expression> ")"
                   | "square" | "rhomb" | "cross" | "plus" | "dot" | "microdot"

<color function> ::= "rgb(" <expression>, <expression>, <expression> ")"
                   | "rainbow(" <expression> ")"
                   | "black" | "blue" | "cyan" | "darkGray" | "gray" | "green"
                   | "lightGray" | "magenta" | "orange" | "pink" | "red" | "white"
                   | "yellow"
```